WHAT IS CLAIMED IS:

1. A method of accessing a plurality of memories in an interleaved manner using a contiguous logical address space, the method comprising:

providing at least one map table, the at least one map table including a plurality of entries, each entry including a plurality of entry items, each entry item identifying one of the memories;

receiving a first logical address, the first logical address including a plurality of address bits, the plurality of address bits including a first set of address bits corresponding to a first set of entries in the at least one map table;

identifying a first entry in the first set of entries based on the first set and a second set of the address bits;

identifying a first entry item in the first entry based on a third set of the address bits; and

accessing the memory identified by the first entry item.

2. The method of claim 1, wherein the first, second, and third sets of address bits are non-overlapping.

3. The method of claim 1, wherein the first, second, and third sets of address bits are each separated from one another by a plurality of other bits.

4. The method of claim 1, wherein the first set of address bits include more significant bits than the second set of address bits, and wherein the second set of address bits include more significant bits than the third set of address bits.

5. The method of claim 1, and further comprising:

storing a plurality of memory offset values in the at least one map table;

identifying one of the memory offset values based on the first logical address; and

wherein the memory identified by the first entry item is accessed at a memory location based at least in part on the identified memory offset value.

6. The method of claim 1, wherein the first logical address is a processor address

7. The method of claim 1, wherein the at least one map table is organized into a plurality of rows and a plurality of columns, and wherein each row corresponds to one of the plurality of entries and each column within a row corresponds to one of the plurality of entry items.

8. The method of claim 1, and further comprising:
providing a multi-bit mask value;
providing a plurality of multi-bit match values;
extracting the first set of address bits from the first logical address using the multi-bit mask value; and
comparing the extracted first set of address bits to the plurality of multi-bit match values to identify a match.

9. The method of claim 1, and further comprising:
providing at least one multi-bit mask value;
providing a plurality of multi-bit match values;
extracting the second set of address bits from the first logical address using the at least one multi-bit mask value;
comparing the extracted second set of address bits to the plurality of multi-bit match values; and
wherein the first entry in the first set of entries is identified based at least in part on the comparison of the extracted second set of address bits to the plurality of multi-bit match values.

10. The method of claim 1, and further comprising:
providing a plurality of multi-bit mask values;
providing a plurality of multi-bit match values;

selecting one of the plurality of multi-bit mask values based on a desired interleave entry size;

extracting the second set of address bits from the first logical address using the selected multi-bit mask value;

comparing the extracted second set of address bits to the plurality of multi-bit match values; and

wherein the first entry in the first set of entries is identified based at least in part on the comparison of the extracted second set of address bits to the plurality of multi-bit match values.

11. The method of claim 1, wherein the memories each include at least one memory segment, the memory segments organized into groups, the memory segments in each group having a uniform size, and wherein each entry in the at least one map table corresponds to one of the groups of memory segments.

12. A method of mapping a contiguous logical address space to a plurality of memories in an interleaved manner, each logical address in the logical address space including first, second, and third non-overlapping address portions, the method comprising:

providing at least one map table, the at least one map table including a plurality of interleave entries, each interleave entry including M entry items representing M-way interleaving of the memories, wherein M is an integer value, and wherein each entry item identifies one of the memories;

associating a plurality of logical addresses with each interleave entry, wherein the logical addresses associated with each interleave entry in a first set of the interleave entries include a common value for the first address portion, the first set of interleave entries including N interleave entries, wherein N is an integer value, and wherein the first set of interleave entries collectively represents a single M x N way interleave entry; and

indexing entry items within the M x N way interleave entry based on the second and the third address portions of the logical addresses associated with the interleave entries in the first set.

13. The method of claim 12, wherein the first, second, and third address portions are separated from one another by a plurality of bits.

14. The method of claim 12, wherein the first address portion of logical addresses includes address bits that are more significant than address bits of the second address portion, and wherein the second address portion of logical addresses includes address bits that are more significant than address bits of the third address portion.

15. The method of claim 12, and further comprising:

storing a plurality of memory offset values in the at least one map table, each memory offset value associated with one of the entry items.

16. The method of claim 12, wherein the logical addresses are processor addresses.

17. The method of claim 12, wherein the at least one map table is organized into a plurality of rows and a plurality of columns, and wherein each row corresponds to one of the plurality of interleave entries and each column within a row corresponds to one of the plurality of entry items.

18. The method of claim 12, and further comprising:

identifying at least one memory segment in each of the plurality of memories; and

organizing the memory segments into groups, wherein the memory segments in each group have a uniform size, and wherein each interleave entry

in the at least one map table corresponds to one of the groups of memory segments.

19. A system for providing interleaved access to a plurality of memories using a contiguous logical address space, the system comprising:

at least one map table, the at least one map table including a plurality of interleave entries, each interleave entry including a base number of entry items, and wherein each entry item identifies one of the memories;

a controller for receiving logical addresses, each logical address including a first portion corresponding to a set of interleave entries in the at least one map table;

an interleave entry size identifier representing a desired number of ways of interleave, the desired number of ways of interleave being a multiple of the base number; and

wherein the controller is configured to identify an interleave entry in the at least one map table based on the first portion of a received logical address, a second portion of the received logical address, and the interleave entry size identifier, and wherein the controller is configured to identify an entry item in the identified interleave entry based on a third portion of the received logical address.

20. The system of claim 19, wherein the first, second, and third portions of the received logical address are non-overlapping and not directly adjacent to each other.

21. The system of claim 19, wherein the first portion of the received logical address includes address bits that are more significant than address bits of the second portion, and wherein the second portion of the received logical address includes address bits that are more significant than address bits of the third portion.

22. The system of claim 19, wherein the at least one map table includes a plurality of memory offset values, each memory offset value corresponding to one of the entry items, and wherein the controller is configured to identify one of the memory offset values based on the received logical address.

23. The system of claim 19, wherein the received logical address is a processor address

24. The system of claim 19, wherein the at least one map table is organized into a plurality of rows and a plurality of columns, and wherein each row corresponds to one of the plurality of interleave entries and each column within a row corresponds to one of the plurality of entry items.

25. The system of claim 19, wherein the controller is configured to mask the received logical address to extract the first logical address portion, and compare the extracted first logical address portion to a plurality of match values to identify a match.

26. The system of claim 19, wherein the interleave entry size identifier is a mask value, and wherein the controller is configured to mask the received logical address with the mask value to extract the second logical address portion, and compare the extracted second logical address portion to a plurality of match values to identify a match.

27. The system of claim 19, wherein the interleave entry size identifier is a mask value, and wherein the controller is configured to select one of a plurality of mask values based on a desired interleave entry size, mask the received logical address with the selected mask value to extract the second logical address portion, and compare the extracted second logical address portion to a plurality of match values to identify a match.

28. The system of claim 19, wherein each of the memories includes at least one memory segment, the memory segments being organized into groups with the memory segments in each group having a uniform size, and wherein each interleave entry in the at least one map table corresponds to one of the groups of memory segments.